Aalto University

Using logical form encodings for unsupervised linguistic transformation

Tommi Gröndahl
N. Asokan

Idea

Linguistic transformation

Linguistic transformation

Controlled change of some property of a sentence, retaining others

- Grammatical: *A dog chased a cat.* → *Did a dog chase a cat?*
- Lexical: *A dog chased a cat.* → *A dog chased a mouse.*

Linguistic transformation

Rule-based [1 - 3]

- Manual work
- Scales badly to novel data
- No training data
- Controllable

Sequence-to-sequence (seq2seq) [4, 5]

- Less manual work
- Better scalability
- Requires a parallel corpus
- Less user control

[1] Afroza Ahmed and King Ip Lin. 2014. Negative Sentence Generation by Using Semantic Heuristics. In The 52nd Annual ACM Southeast Conference (ACMSE 2014).
[2] Jorge Baptista, Sandra Lourenco, and Nuno Mamede. 2016. Automatic generation of exercises on passive transformation in Portuguese. In IEEE Congress on Evolutionary Computation (CEC), pages 4965–4972.
[3] Yonatan Bilu, Daniel Hershcovich, and Noam Slonim. 2015. Automatic claim negation: why, how and when. In Proceedings of the 2nd Workshop on Argumentation Mining, pages 84–93.
[4] Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. Paraphrase Generation with Deep Reinforcement Learning. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 3865–3878, 2018.
[5] Nitin Madnani and Bonnie Dorr. Generating phrasal and sentential paraphrases: A survey of data-driven methods. Journal of Computational Linguistics, 36(3):341–387, 2010.

Linguistic transformation

Combining *n* transformations requires *n* models

Source $\longrightarrow$ Transf(1) $\longrightarrow$ (...) $\longrightarrow$ Transf(n)

John saw Mary $\longrightarrow$ *John didn't see Mary* $\longrightarrow$ *Didn't John not see Mary?*

LF2seq

LF2seq

A novel method for linguistic transformation

- Hybrid approach: combines formal semantics with data-driven methods [6 - 8]
- Rule-based pre-processing + seq2seq
- Unsupervised: no parallel corpus or labelled data
- Allows user-controllable transformations in any combination

[6] I. Beltagy, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J. Mooney. 2016. Representing meaning with a combination of logical and distributional models. The special issue of Computational Linguistics on Formal Distributional Semantics, 42(4):763–808.

[7] D. Garrette, K. Erk, and R. Mooney. 2011. Integrating logical representations with probabilistic information using markov logic. In Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011), pages 105–114.

[8] Mike Lewis and Mark Steedman. 2013. Combined distributional and logical semantics. Transactions of the Association for Computational Linguistics, 1:179–192.

LF2seq

Pipeline

1. Parsing the source sentence
2. Producing LF: approximative "logical form" + grammatical markers
3. LF as input to an encoder-decoder network
4. Decoded target sentence

$$\text{sentence} \xrightarrow{\text{parser + LF-rules}} \text{LF-representation} \xrightarrow{\text{encoder}} \text{LF-encoding} \xrightarrow{\text{decoder}} \text{sentence*}$$

Structure of LF

Structure of the LF-representation

Basis: **dependency grammar** [9]

- Dependency parses can be mapped to LFs [10, 11].
- We build the LF directly from the parse, using no lexical knowledge base.
- Argument structure is mapped directly from syntax, not verb-specific "thematic grids".

Three requirements for LF:

1. Represents semantic information
2. Allows control of grammatical/lexical properties for producing transformations
3. Has an appropriate format to function as encoder input

[9] Louis Tesnière. 1959. Èléments de syntaxe structurale. Klincksieck, Paris.
[10] Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. volume 4, pages 127–140.
[11] Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 89–101. Association for Computational Linguistics.

Thematic structure

Basic thematic structure

- Transitive sentences represented by the triplet <Event, Agent, Theme>
- Intransitives lack Theme: <Event, Agent, Ø>
- Passives can lack Agent: <Event, Ø, Theme>

A dog chases a cat

<chase, dog, cat>

A dog jumps

<jump, dog, Ø>

A cat is chased

<chase, Ø, cat>

Thematic structure

Modifiers

- Most modifiers (e.g. adjectives/adverbs) are additional predicates of the Event/Agent/Theme.
- Possessors are nouns with a separate possessive marker
- Modification is conjunctive, in line with (Neo-)Davidsonian semantics [12 - 15].

A brown dog chased a black cat yesterday

<chase, dog, cat>

<yesterday, brown, black>

Mary's dog chases John's cat

<chase, dog, cat>

<∅, Mary-POSS, John-POSS>

[12] Donald Davidson. 1967. The logical form of action sentences. In N. Resher, editor, The Logic of Decision and Action, pages 81–95. University of Pittsburgh Press, Pittsburgh.
[13] James Higginbotham. 1985. On semantics. Linguistic Inquiry, 16:547–593.
[14] Paul Pietroski. 2005. Events and Semantic Architecture. Oxford University Press, Oxford.
[15] Paul Pietroski. 2018. Conjoining Meanings: Semantics without Truth-values. Oxford University Press, Oxford.

Thematic structure

Modifiers

- Prepositional phrases are analogical to transitive verbs [16].
- Prepositions are marked as modifying the Event/Agent/Theme position.

A dog with a tail runs in the yard

<run, dog, Ø>

<in, EVENT, yard>

<with, AGENT, tail>

[16] Kenneth Hale and Samuel Jay Keyser. 2002. Prolegomenon to a Theory of Argument Structure. MIT Press, Cambridge.

Thematic structure

Relative pronouns

- Treated as normal words, except marked as coreferential to a prior Agent/Theme position.
- If marked for neither Agent not Theme, modifies an Event by default.

A dog that has fleas chases a man who owns a cat, which is funny

<chase, dog, man>

<have, that-AGENT, fleas>

<own, who-THEME, cat>

<be, which, funny>

Grammatical features

- Verbal/clausal:
  - Force: declarative/imperative/question
  - Truth: affirmed/negated
  - Voice: active/passive
  - Tense: past/present/perfect/pluperfect
  - Aspect: perfective/imperfective

- Nominal:
  - Number: singular/plural
  - Definiteness: definite/indefinite
  - Possessive: possessive/non-possessive

- Adjectival/adverbial:
  - Comparison class: comparative/superlative

Grammatical features

- Grammatical features are represented as Boolean features (0/1).
- Nominal features are separated between Agent and Theme.
- Adjectival/adverbial features are separated between Event, Agent, and Theme.
- Relative pronouns are marked for coreference with a prior Agent/Theme.
- Prepositions are marked for modifying the Event/Agent/Theme.

Grammatical features

Feature class	Index	Feature meaning
Force	1	Question
	2	Imperative
Truth	3	True
Voice	4	Active
	5	Passive
Tense	6	Present
	7	Past
	8	Perfect
Aspect	9	Imperfective
Preposition	10	Preposition modifying Event
	11	Preposition modifying Agent
	12	Preposition modifying Theme
Noun	13	Agent is possessive
	14	Agent is plural
	15	Agent is definite
	16	Agent is an Agent-RP
	17	Agent is a Theme-RP
	18	Theme is possessive
	19	Theme is plural
	20	Theme is definite
	21	Theme is an Agent-RP
	22	Theme is a Theme-RP
Adjective/adverb	23	Event is comparative
	24	Event is superlative
	25	Agent is comparative
	26	Agent is superlative
	27	Theme is comparative
	28	Theme is superlative

LF

LF

An LF-vector has two parts

- Boolean grammatical features

A dog chased a cat

< 0, 0, 1, …, 0 >

LF

An LF-vector has two parts

- Boolean grammatical features
- Word triplet for thematic structure

A dog chased a cat

< 0, 0, 1, …, 0, chase, dog, cat >

LF

An LF-vector has two parts

- Boolean grammatical features
- Word triplet for thematic structure (pre-trained word embeddings)

A dog chased a cat

< 0, 0, 1, …, 0, chase, dog, cat >

(0.3, 0.2, ...) (-0.1, 0.4, ...) (0.7, -0.6, ...)

LF

An LF-vector has two parts

- Boolean grammatical features
- Word triplet for thematic structure (pre-trained word embeddings)
- Zero vector used as empty token embedding

A dog ran

< 0, 0, 1, …, 0, run, dog, ø>

(0.5, -0.1, ...) (-0.1, 0.4, ...) (0.0, 0.0, ...)

LF

The LF of a whole sentence is a sequence of LF-vectors

- Interpreted as the conjunction of the interpretations of each LF-vector.
  - Idealization; a more complex mapping needed for non-entailing modifiers, intensional contexts etc.

The first LF-vector in the sequence always contains the main verb as the Event.

- Relevant for transformations, which need to target the main verb.
- (For current purposes, we discard sentences without a main verb.)

A brown dog chased a black cat yesterday

< 0, 1, …, 0, chase, dog, cat >

< 0, 1, …, 0, yesterday, brown, black >

LF2seq: LFs to English

Training

Training

The decoder is trained to reproduce the original sentence.

$$\text{sentence} \xrightarrow{\text{parser + LF-rules}} \text{LF-representation} \xrightarrow{\text{encoder}} \text{LF-encoding} \xrightarrow{\text{decoder}} \text{sentence*}$$

minimize error

Using LF2seq for transformation

Using LF2seq for transformation

Grammatical transformation

- Changing grammatical features in the LF

< 0, 1, …, 0, chase, dog, cat > → A dog chased a cat

< 1, 1, …, 0, chase, dog, cat > → Did a dog chase a cat?

< 0, 0, …, 0, chase, dog, cat > → A dog didn't chase a cat.

Using LF2seq for transformation

Grammatical transformation

- Changing grammatical features in the LF

Lexical transformation

- Changing thematic arguments in the LF

< 0, 1, …, 0, chase, dog, cat > $\longrightarrow$ A dog chased a cat

< 0, 1, …, 0, chase, dog, bird > $\longrightarrow$ A dog chased a bird

Using LF2seq for transformation

Grammatical transformation

- Changing grammatical features in the LF

Lexical transformation

- Changing thematic arguments in the LF

Free combination is possible

< 0, 1, …, 0, chase, dog, cat >	→	A dog chased a cat
< 1, 0, …, 0, chase, dog, cat >	→	Didn't a dog chase a cat?
< 1, 0, …, 0, chase, dog, bird >	→	Didn't a dog chase a bird?

Possible applications

Grammatical or lexical transformations

Pedagogical applications

- Automatic generation of exercises for language learning
- Grammaticality checking on a syntactic level (e.g. agreement)

Automatic paraphrasing for e.g. style transfer

- Grammatical variants with little semantic impact: e.g. active/passive
- Synonym replacement on a lemma-level (e.g. WordNet) with automatic inflection

Question answering/generation

- Finding a declarative sentence corresponding to a question, or vice versa
- Chat-bots, automatic customer service

Data augmentation

- Generating artificial data to increase the training set for machine learning

Text generation

Generating LFs directly instead of parsing an input sentence

$$\text{sentence} \xrightarrow{\text{parser + LF-rules}} \text{LF-representation} \xrightarrow{\text{encoder}} \text{LF-encoding} \xrightarrow{\text{decoder}} \text{sentence*}$$

Text generation

Generating LFs directly instead of parsing an input sentence

$$??? \xrightarrow{\text{LF-rules}} \text{LF-representation} \xrightarrow{\text{encoder}} \text{LF-encoding} \xrightarrow{\text{decoder}} \text{sentence*}$$

Machine translation (limited)

Requirements

- Lemma-level dictionary between the source language (SL) and the target language (TL)
- SL-parser
- TL-parser
- LF-rules for both parsers

Restricted as a stand-alone solution, but potentially helpful in unsupervised MT [17 - 19]**.**

[17] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. CoRR, abs/1710.11041.
[18] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. In International Conference on Learning Representations (ICLR)
[19] Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato.]2018. Phrase-based & neural unsupervised machine translation. CoRR, abs/1804.07755

Experimental results

Implementation

LF

- Dependency parsing: Spacy
- Word embeddings: 300-dimensional Glove vectors trained on a Common Crawl corpus
- Consequently, LF-vectors were of size 3 x 300 + 28 = 928
- Max LF-sequence length: 10

Training data

- 8.5 million English sentences derived from multiple corpora (max length = 20 words):
  - Stanford parallel corpora
  - Tatoeba
  - OpenSubtitles 2018
  - SNLI
  - SICK
  - Aristo-mini
  - Example sentences from WordNet
- 50 000 sentences for validation

Implementation

Encoder-decoder network

- Encoder and decoder: two-layer LSTMs with 600 hidden units in each layer
- Attention [20] of size 9 used in the decoder.
- Batch size 128
- Negative log-likelihood loss
- Adam optimizer [21]
- Learning rate 0.001 until epoch 5, 0.0001 in epochs 5-7
- Validation loss no longer decreased after epoch 7: these weights used for the test

[20] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473.
[21] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In International Conference on Learning Representations.

Sentence reproduction

Sentence reproduction

Measuring proximity to the original sentence (BLEU)

- 1000 random test sentences

Sentence reproduction

Measuring proximity to the original sentence (BLEU)

- 1000 random test sentences
- BLEU score: 74.45

Grammatical transformation

Grammatical transformation

Transforming a sentence to a different grammatical class

- 300 single-clause test sentences from different classes
- 14 transformation directions

Manual evaluation on 50 transformations from each direction

Automatic evaluation with back-transformation

- Sentence transformed into the target class.
- Result transformed back to the original class
- BLEU score measured with the original

Grammatical transformation

Results

- 66 % of manually evaluated transformations were perfect
- Back-transformation BLEU scores between 55.29 - 81.82

Grammatical transformation: automatic evaluation

Transformation	Correct target category	Back-transformation	
		Identical	BLEU
Declarative-question	299/300	136/299	64.52
Declarative-question	299/300	136/299	81.82
Affirmed-negated	300/300	140/300	67.95
Negated-affirmed	300/300	164/300	79.38
Active-passive	300/300	102/300	55.29
Passive-active	299/300	83/299	55.55
Present-past	300/300	164/300	75.19
Past-present	300/300	148/300	70.47
Present-perfect	300/300	157/300	72.92
Perfect-present	299/300	159/299	76.92
Present-pluperfect	300/300	154/300	73.01
Pluperfect-present	300/300	122/300	67.28
Perfective-imperfective	300/300	151/300	68.28
Imperfective-perfective	300/300	155/300	74.51

Grammatical transformation: manual evaluation

Transformation	Perfect	Grammatical errors		Lexical errors	
		Target class	Elsewhere	Target class	Elsewhere
Declarative–question	35	3	7	1	10
Question–declarative	37	4	5	0	7
Affirmed–negated	27	4	9	0	7
Negated–affirmed	43	2	5	1	4
Active–passive	24	10	12	8	16
Passive–active	22	11	12	7	22
Present–past	38	0	4	0	9
Past–present	34	1	8	0	10
Present–perfect	34	2	7	2	11
Perfect–present	37	0	5	1	12
Present–pluperfect	35	2	7	1	8
Pluperfect–present	27	5	7	2	16
Perfective–imperfective	32	7	6	2	11
Imperfective–perfective	38	2	7	2	9

Summary

- We developed LF2seq: a method to enact linguistic transformations in any combination.
- The model is unsupervised: trained on a monolingual corpus
- We receive high BLEU scores in both sentence reproduction and back-transformation.
- We discussed potential applications of LF2seq, ranging from paraphrasing to text generation.

Questions?

https://ssg.aalto.fi/research/projects/deception-detection-via-text-analysis/